Validation of the PCN Concept; Mobility, Traffic Flow Confidentiality and Protection Against Directed Attacks

Per Carlén
Contractor for: FMV – Swedish Defence Materiel Administration
Peldakon
Rickebasta 219, SE-74192 Knivsta
SWEDEN

pc@peldakon.se

ABSTRACT

Protected Core Networking (PCN) is an approach that aims to provide a highly flexible networking environment that, even in the case of directed attacks against the communications infrastructure, ensures continued operation of critical communications.

To verify that the PCN concept provides the expected benefits, a proof-of-concept prototype was developed.

This paper will focus on three areas from the prototyping work; Mobility of coloured clouds, mechanisms for Traffic Flow Confidentiality (TFC) and the capability of a network built on PCN principles to withstand directed attacks.

Coloured clouds (CCs) are the users in the network, typically being a part of an information infrastructure confidentiality-protected by an IP-crypto. An important aspect, in Network Enabled Capability environments, is flexibility in terms of mobility of the CCs.

An analysis of encrypted traffic flows, looking at sizes and intervals of packets etc, between CCs can reveal important information of the communication like type of traffic and chain-of-command. TFC-mechanisms provide measures against analysis of traffic flows.

Directed attacks on a network with the intention to lower the capacity of the infrastructure, thereby hindering important information to reach its destination, is addressed by ensuring that only authorized entities can send traffic at a pre-agreed maximum rate on the network.

This paper will describe how the functionality was implemented in a prototype and further present the results from an experiment where the prototype was used in a simulated operational setting.

1.0 INTRODUCTION

The PCN-concept was developed to allow a federated network to provide required communication-services for operations with high agility. The prototyping work was performed in coordination with the conceptual work and both activities were part of a NATO Research and Technology Organisation (RTO) Task Group on PCN that ended on December 31st 2009.

Even though the prototype integrates functionality with quite a few technologies, this paper doesn't describe the entire prototype but is rather focusing on the three earlier mentioned topics.

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **NOV 2010**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**

4. TITLE AND SUBTITLE
Validation of the PCN Concept; Mobility, Traffic Flow Confidentiality and Protection Against Directed Attacks

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
FMV Swedish Defence Materiel Administration Peldakon Rickebasta 219, SE-74192 Knivsta SWEDEN

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR’S ACRONYM(S)

11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADA564697. Information Assurance and Cyber Defence (Assurance de l’information et cyberdefense). RTO-MP-IST-091

14. ABSTRACT

Protected Core Networking (PCN) is an approach that aims to provide a highly flexible networking environment that, even in the case of directed attacks against the communications infrastructure, ensures continued operation of critical communications. To verify that the PCN concept provides the expected benefits, a proof-of-concept prototype was developed. This paper will focus on three areas from the prototyping work; Mobility of coloured clouds, mechanisms for Traffic Flow Confidentiality (TFC) and the capability of a network built on PCN principles to withstand directed attacks. Coloured clouds (CCs) are the users in the network, typically being a part of an information infrastructure confidentiality-protected by an IP-crypto. An important aspect, in Network Enabled Capability environments, is flexibility in terms of mobility of the CCs. An analysis of encrypted traffic flows, looking at sizes and intervals of packets etc, between CCs can reveal important information of the communication like type of traffic and chain-of-command. TFC-mechanisms provide measures against analysis of traffic flows. Directed attacks on a network with the intention to lower the capacity of the infrastructure, thereby hindering important information to reach its destination, is addressed by ensuring that only authorized entities can send traffic at a pre-agreed maximum rate on the network. This paper will describe how the functionality was implemented in a prototype and further present the results from an experiment where the prototype was used in a simulated operational setting.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT **SAR**	18. NUMBER OF PAGES **24**	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**			

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

1.1 PCN Concept, Entities and Interfaces

In PCN the network infrastructure is built using routers with enforcement-functionality, these routers are referred to as E-nodes. A set of E-nodes form a Protected Core Segment (PCS) often under a common authority. A Protected Core (PCore) is a set of PCSs, building a federated infrastructure to be used by the CCs. The CCs are the users in PCN and they use the PCore to transport information to other CCs.

There are two types of interfaces defined, in the concept, for interconnection; PCN-1 which is what PCSs use when they connect and PCN-2 which is what a CC uses when connecting to the PCore.

1.2 Putting the Prototype Together

Since a lot of PCN-functionality didn’t exist commercially or as open-source, a flexible base with the possibility to add integrated, self-developed, functions was needed. The choice after some testing, mainly on unicast- and multicast-routing, fell upon Ubuntu-server 8.04.1 [20].

Development of special functionality, which will be further described in following chapters, was done using C programming language in Eclipse IDE (Integrated Development Environment) [18].

The prototype included three different types of machines; E-nodes, Z-devices and CC-servers.

As shown in figure 1, the E-nodes build the PCore used by the CCs. A CC in the prototype consists of a Z-device, being an IP-crypto, at the edge connecting to the PCore and a CC-server sitting behind the Z-device.

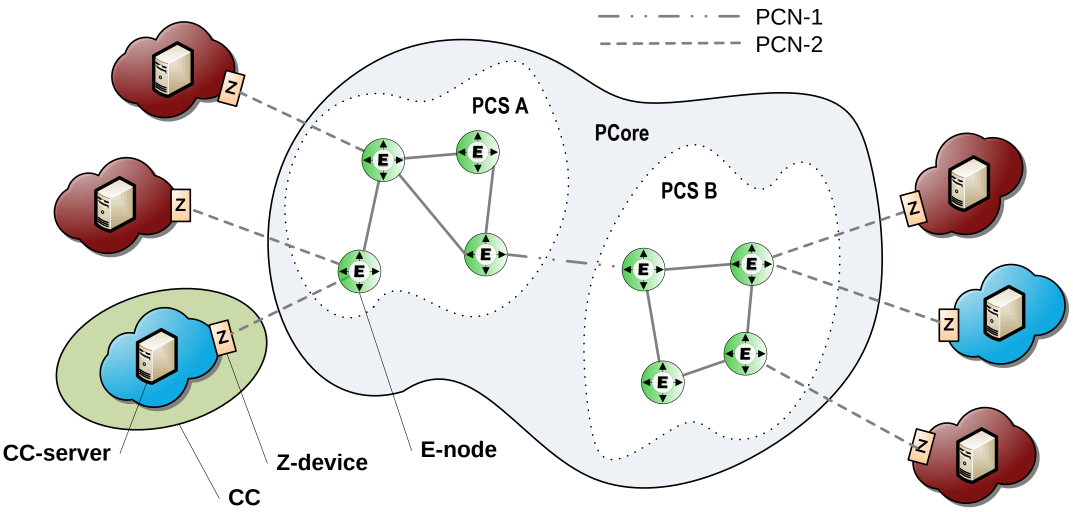


Figure 1: Different entities in the PCN prototype

Virtualization technology, using VMWare ESXi [19], enabled easy scaling and sharing of template-machines within the task group during the experimentation. The prototype was distributed in the sense that five organizations in different nations put up their own PCS(s) and interconnected, forming a PCore, over IPv6 Internet.

1.2.1 E-node

The E-node is first of all capable of routing and forwarding packets.

Unicast-routing is done with Quagga [1], OSPF is used within the segments and BGP between segments. Quagga is an open-source routing software suite that has support for IPv6, OSPFv3 [2] and BGPv4 [3].

Multicast-routing is done with mrd6 [4], using PIM [5]. Mrd6 is also used for MLD [6], the equivalent in IPv6 to IGMP. Since there really is no scalable solution available for inter-domain multicast-routing, even though there is a standard – BGMP [7], PIM was used both within and between PCSs.

The E-node, in the prototype, can handle three different types of interfaces; internal (connecting to other E-nodes), PCN-2 and non-PCN. Most effort was put on the PCN-2 interface, because the focus was decided to lie on the capability delivered to the actual users of the PCore.

Apart from routing, the following was added as PCN-functionality and implements the PCN-2 interface:

- Certificate-based authentication using a subset from EAP-TLS [8,9,10] (app developed by the author)
- Authorization based on information in the certificate used in authentication (app developed by the author)
- Link protection using MAC Security (802.1AE - layer2 encryption) from SafeNet[21] and ip6tables (Linux firewall for IPv6)
- Traffic volume confidentiality using a padding-implementation (app developed by the author)
- Service Level Agreement (SLA)- negotiation, policing and shaping based on it (app developed by the author)
- Traffic rate monitoring (app developed by the author)

For the internal interfaces all of the above but SLA-negotiation was implemented, and for non-PCN interfaces only policing and shaping was implemented.

1.2.2 Z-device

The Z-device is capable of connecting to an E-node through a PCN-2 interface. Once connected, the Z-device uses a mechanism for peer-discovery to discover and set up IPSec tunnels to clouds of the same colour.

The following has been added as PCN-functionality:

- Certificate-based authentication using a subset from EAP-TLS (app developed by the author)
- Authorization based on information in the certificate used in authentication (app developed by the author)
- Link protection using MACSec (802.1AE layer2 encryption) from SafeNet and ip6tables
- Traffic volume confidentiality using a padding-implementation (app developed by the author)
- SLA-negotiation, policing and shaping based on it (app developed by the author)

A special implementation for IPSec peer discovery, using multicast, was also developed (by the author).

1.2.3 CC-server

The CC-server is merely used for CC monitoring purposes. Without it no IPSec tunnels would be established since no traffic would trigger the IPSec setup.

1.3 Experiment Setup

To further investigate the operational benefits of using a network with PCN principles, the prototype was used in a simulated operational experiment. The CWID 2009 Land Component Command (LCC) scenario was used as a basis. Since it only included the mobility parts, events were added to further address other functionality in the prototype such as TFC and the capability to withstand directed attacks.

As shown in figure 2 below, five nations hosting seven PCSs and 19 CCs (four different colours) participated in the experiment. The nations were interconnected with IPSec tunnels over IPv6 Internet.

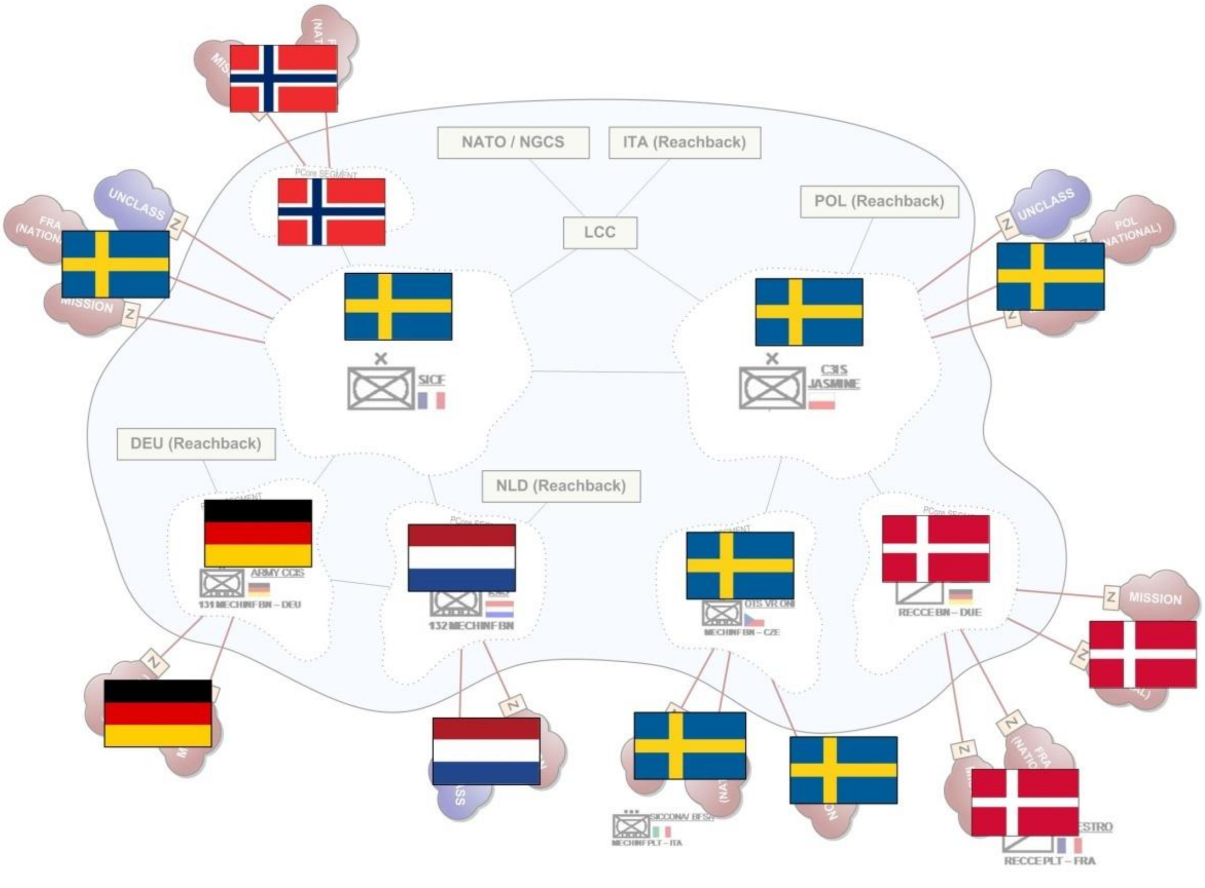


Figure 2 - Implementation of the operational scenario in the PCN prototype

2.0 MOBILITY OF COLOURED CLOUDS

To provide for flexibility and timeliness in military operations, a relocation of a coloured cloud should not require any manual reconfiguration.

2.1 PCN Concept

In PCN this is achieved by auto-configuration of coloured cloud-address in the connection-phase and then a following discovery of the peers. The connection-phase also includes authentication, authorization, SLA-negotiation and setting up protection of the link. A trust relationship between the nation-CAs has to be in place to enable CCs connecting to other PCSs than its own national PCS.

2.2 Implementation

When a CC connects it first performs a mutual authentication with the PCore. If a CC from one nation connects to a PCS belonging to another nation, the two connecting devices (a z-device and an E-node) need to trust each other. This was achieved by having the other parties CA certificate installed locally as a trusted CA certificate. An authorization to use the network based on the identity of the coloured cloud follows and then there is SLA-negotiation stating a coloured cloud usage of services, offered by the network. An optional setup of link protection, described in chapter 3, follows. Then the CC gets an IPv6 address by auto-configuration. After these initial steps, the connected CC discovers its peers with a multicast protocol. This in turn makes it possible for the CC to go ahead and set up security associations with its peers. This chapter will focus mainly on auto-configuration and IPSec peer discovery.

2.2.1 Auto-Configuration

In the prototype, stateless auto-configuration [11] is used with radvd (router advertisement daemon). Radvd is open-source software that implements advertisements of IPv6 prefixes and router addresses thus enabling a CC to get an address automatically once connected.

2.2.2 IPSec Peer Discovery

The peer discovery method is implemented using multicast and Linux ipsec-tools, the preferred solution would have been using an approach with group keying [12]. However a more simple solution was chosen due to time restrictions.

When a peer has connected and received and address, it sends a multicast packet to a colour(information domain)-specific multicast-group. Peers, of the same colour, listening for packets on the specific multicast-address, receive packets from the new peer and answers with unicast. The new peer gets the unicast-packets and the peer discovery is complete. Information regarding the CC-internal, also referred to as red network, addresses is also included in the peer discovery.

Discovered peers and red networks are added to setkey (ipsec-tools). When traffic from inside a CC matches a red network of a peer, racoon (IKE-daemon) will start authenticating using certificates. After a successful authentication, IPSec security associations will be set up when triggered by matching traffic, the tunnel is established and traffic can flow encrypted between the CCs.

In the prototype, all CCs of one colour had credentials from one CA.

Racoon and ipsec-tools are open-source software, implementing IKE and IPSec.

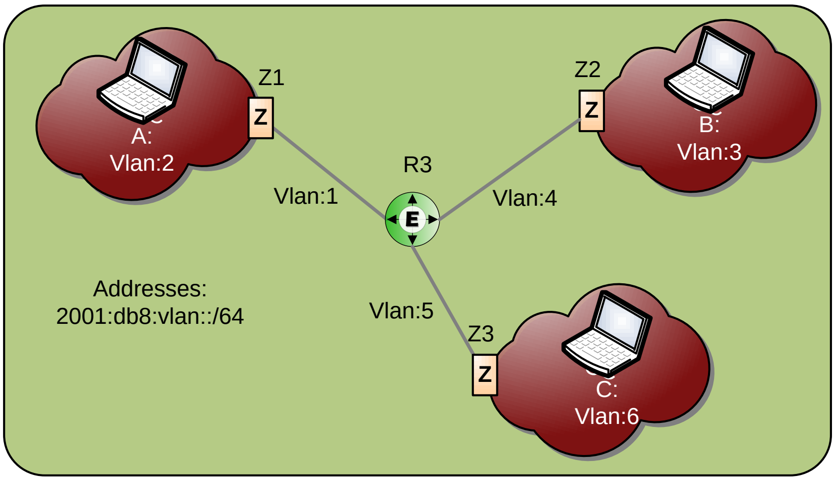


Figure 3: IPSec peer discovery in the PCN prototype

Following example, shown in figure 3, describes what happens at C when it connects:

- A and B are already connected, sending MLD join on colour-specific multicast-group
- C connects, gets address 2001:db8:5::10
- C flushes all ipsec-peers
- C sends multicasts on colour-specific multicast-group for 10 seconds and then goes into response mode – sending MLD join
- Upon receipt of unicast from A&B, their addresses and red network addresses are stored in discovered-peers
- After sending multicasts, C compares ipsec-peers & discovered-peers
- C adds A to ipsec-peers (peer: 2001:db8:1::10, red network: 2001:db8:2::/64)
- C adds B to ipsec-peers (peer: 2001:db8:4::10, red network: 2001:db8:3::/64)
- When traffic from within C is sent to A, this triggers IKE and a following tunnel-setup.

What happens at A:

- A and B are connected, sending MLD join
- C connects
- A gets multicast from C (2001:db8:5::10) and responds with unicast packet
- A checks ipsec peers and removes C if it exists
- A adds C to ipsec-peers (peer: 2001:db8:5::10, red network: 2001:db8:6::/64)

The solution was good enough for the prototype - tuning the discovery-mechanism would most likely lead to faster discovery (about 1 minute at the time writing).

2.3 Experiment

The experiment on mobility was divided in two runs with the following objectives;

- Run 1: Show the automatic network configuration capability of PCN in a federated environment.
- Run 2: Show the power of a PCore in a federated setting when changes occur in the operational structure (a unit passed operational control to another HQ), by quick reconfiguration of the network.

2.3.1 Setup

Figure 4, below, shows the dynamics going from a start position through two runs.

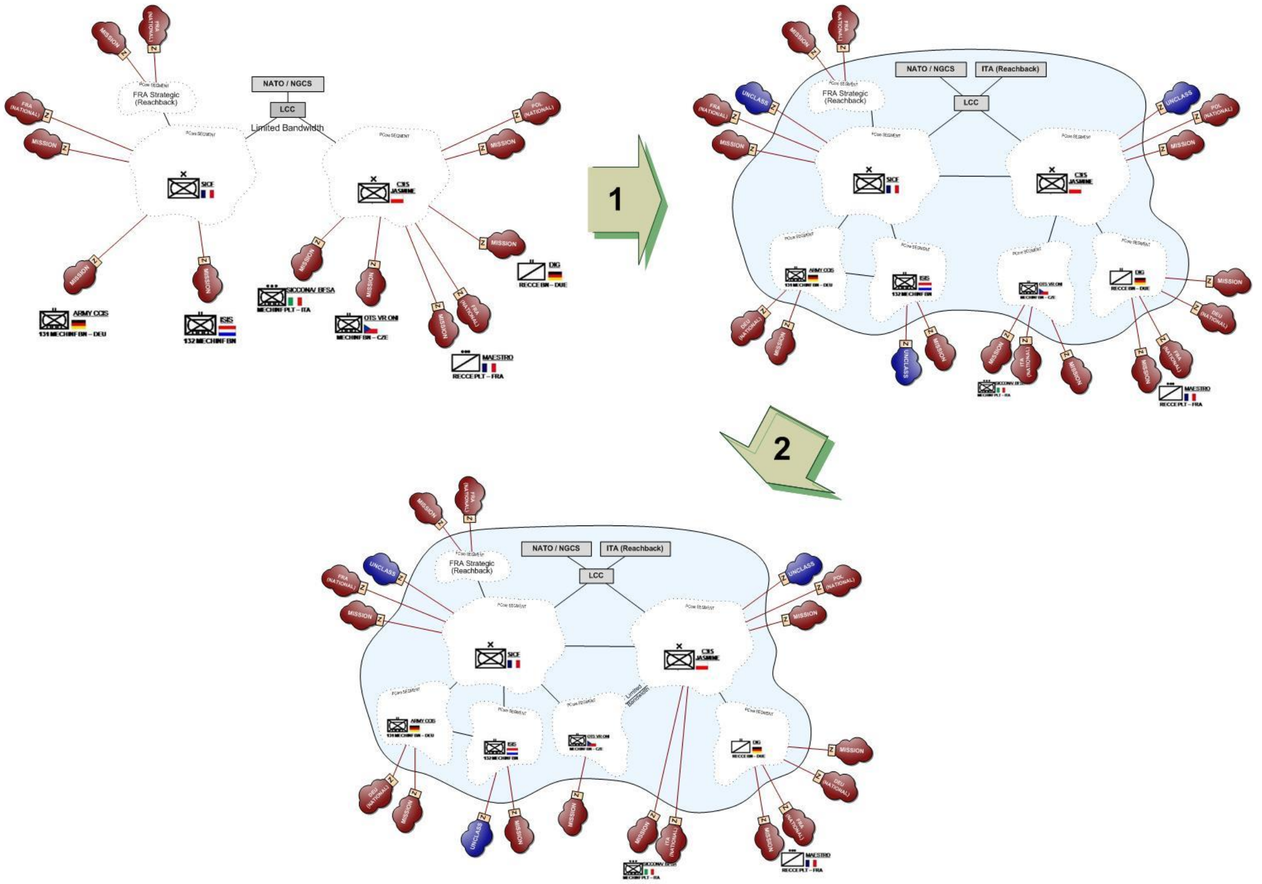


Figure 4: Relocation of CCs and PCSs during the experiment

Relocation of a CC was performed by disconnecting a virtual machine, hosting a CC with its identity, at one location and connecting a copy of the virtual machine at another. CCs with the same identity and same red network, could therefore exist at several geographically locations – though only one instance connected at a time.

2.3.2 Results

In run 1, most CCs (8/9) discovered each other and were able to communicate with each other within about two minutes. It's hard to analyze actual time for discovery since CCs were connected sequentially. It should also be mentioned that the mechanism for peer-discovery was not developed for performance, but all peers should have at least discovered each other within one minute. The mechanism for peer discovery relies on multicast, problems encountered during the runs were due to misconfigurations like forwarding disabled in one E-node and BGP advertisements leading to flapping (routes that come and go leading to instability).

It also turned out that mrd6 (multicast routing daemon) was not very stable when all seven PCSs with their E-nodes were connected, mrd6 had to be restarted for the multicast-tree to be successfully built.

Seamless mobility is not only about finding the peers, it also includes the entire connection-phase at PCN-2; authentication->authorization->link protection setup(optional)->address assignment->SLA negotiation. In the experiment, the connection-phase lasted about 15 seconds without MACSec and 30 seconds with MACSec.

In run 2, the connection between the segments showed no real issues, routing was done with BGP and worked within a minute from PCSs being connected.

2.3.3 Conclusions

IPSec peer discovery and auto-configuration work well and proves to be a good concept for mobility without any need for manual reconfiguration. The encountered problems were all related to multicast- and underlying unicastrouting, due to misconfigurations and unstable routing-software.

To successfully do multicast in a federated environment, a good design for both intra- and inter-domain multicast is of big importance. The experiment also shows that there is a real need for an implementation of an intra-domain multicast protocol for IPv6, something that doesn’t exist today. Running PIM throughout an entire PCore does not seem like a good solution, although it may work in small networks.

Relocation of segments showed no real issues, routing was done with BGP and worked within a minute from being connected.

3.0 TFC

When data is transmitted over a network, protocols with specific headers are added. On Ethernet, information is transferred in frames of variable sizes and intervals. Even though using IPSec, a statistical analysis of sizes and intervals of packets can be performed revealing important information of the communication [13].

There are some mechanisms that can make it more difficult for doing such an analysis, thus providing TFC. By using encryption at lower layers, address and service hiding can be achieved. Hiding sizes and intervals – volume confidentiality – can be provided by sending fixed size frames at a constant rate [14].

3.1 PCN Concept

In PCN the coloured clouds themselves stand for the information confidentiality and the network is offering them TFC as a service, thereby providing measures against traffic flow analysis. In the network there will be TFC-provisioned paths, the coloured clouds will signal its need for TFC and the PCore will route the traffic accordingly.

3.2 Implementation

In the prototype, TFC-mechanisms is built on layer2-encryption together with padding and dummy packet generation (providing both packet-size and traffic-volume confidentiality). By using multipath routing in the PCore, E-nodes can forward traffic on different paths in accordance with the, by a CC, signalled TFC-requirement.

3.2.1 Multipath Routing

Multipath routing is implemented by using dscp-marking of packets and rulebased-routing in iproute2-package. There is currently no integration done with OSPF, which of course would have been preferred, routing based on TFC-level is done manually and is static. Marking of the packets is performed by using ip6tables (Linux firewall for IPv6) and tc (traffic control, part of iproute2).

The implementation is shown in figure 5 below. All traffic from A heading for B will traverse the lower path. Other traffic will take the upper path, which is an unsecure link using Internet (simulated).

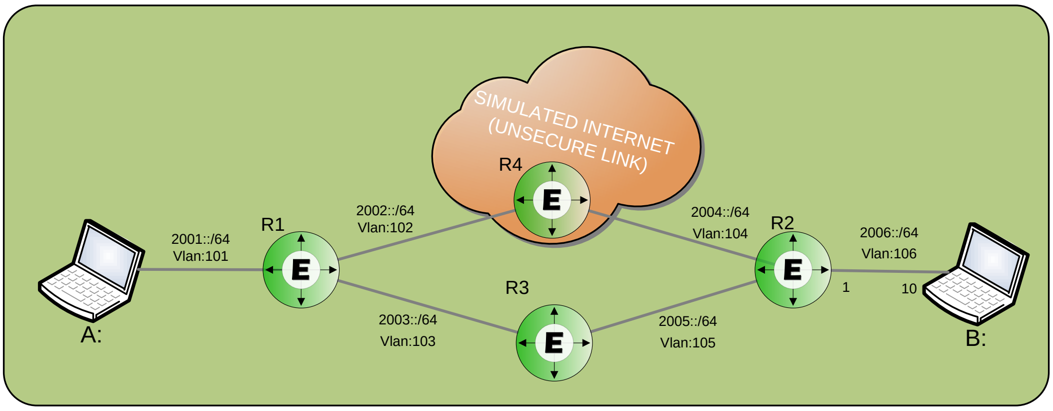


Figure 5: Implementation of multipath routing

Configuration in A is done with ip6tables and tc. Ip6tables will mark the packets heading for 2006::10 internally, which is then used by tc to set one bit in the traffic-control-byte in the IPv6 header.

Configuration in R1: Two routing tables, TFC(lower path) and nonTFC(upper path). Ip6tables marks traffic internally that matches b0(TFC-signalling bit in traffic-control-byte in the IPv6 header). Rules then point to different routing tables depending on this marking.

3.2.2 MACSec

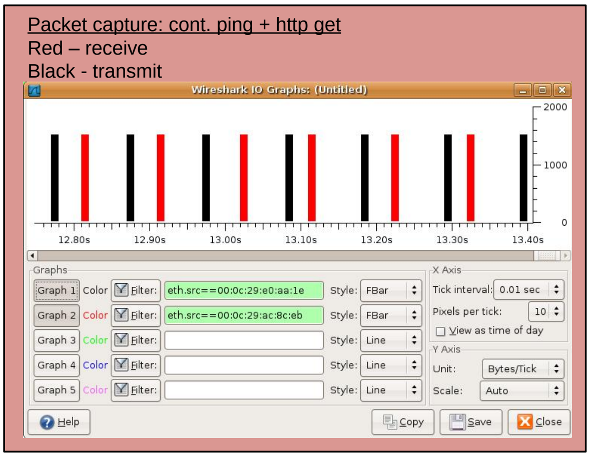
MACSec, also referred to as 802.1AE, is an IEEE standard for layer 2-encryption. By using MACSec, all IPv6 information, both header and payload is hidden thereby providing, among others, address confidentiality. MACSec, like IPSec needs keys to be able to encrypt and decrypt data. For this purpose, parts of 802.1X-2010 (extends 802.1X to support 802.1AE) was used. An agreement with SafeNet during the prototyping-work allowed the usage of QuickSec, which is a development-kit including source-code for 802.1AE and 802.1X-2010 (also known as 802.1XREV).

Having the protocols for authentication and encryption in place, a shared secret was needed.

In the prototype, the connection is authenticated on layer 2 with EAP-TLS (see chapter on authentication later in paper). This protocol will, besides from authenticating, create a shared secret which is used later on for authentication in 802.1XREV.

3.2.3 Padding

One way of providing traffic volume confidentiality is by only sending packets of same size at with a fixed interval as in figure 6.



Packet capture: cont. ping + http get
Red – receive
Black - transmit

Wireshark IO Graphs: (Untitled)

Graphs

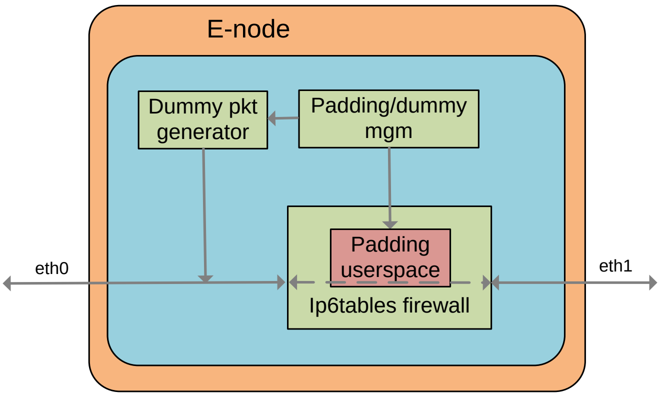
Graph 1	Color	Filter:	eth.src==00:0c:29:e0:aa:1e	Style: FBar
Graph 2	Color	Filter:	eth.src==00:0c:29:ac:8c:eb	Style: FBar
Graph 3	Color	Filter:		Style: Line
Graph 4	Color	Filter:		Style: Line
Graph 5	Color	Filter:		Style: Line

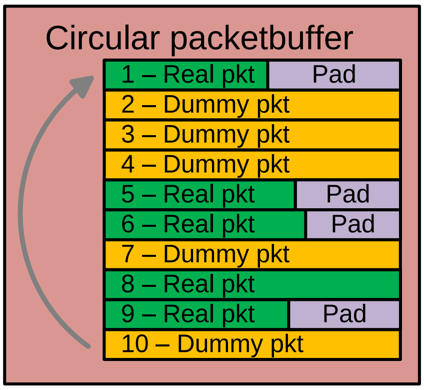
X Axis
Tick interval: 0.01 sec
Pixels per tick: 10
☐ View as time of day

Y Axis
Unit: Bytes/Tick
Scale: Auto

Figure 6: IO graph from Ethernet link utilizing padding and dummy packet generation

This is implemented in the prototype, as shown in figure 7 and 8, by using padding of packets and insertion of dummy packets when there is no real traffic on the link. The application uses ip6tables to queue packets to the “padding”-application.





Circular packetbuffer

1 – Real pkt	Pad
2 – Dummy pkt	
3 – Dummy pkt	
4 – Dummy pkt	
5 – Real pkt	Pad
6 – Real pkt	Pad
7 – Dummy pkt	
8 – Real pkt	
9 – Real pkt	Pad
10 – Dummy pkt	

Figure 7 (left),: Implementation of TFC volume confidentiality
Figure 8 (right): Circular packet-buffer used by padding-application

The userspace padding/dummy-packet generator uses a circular packetbuffer (figure 8) and is doing the following:

- Pads every ipv6 packet up to 1500 bytes
- Generates dummy packets
- Inserts real packets in circular packet buffer queue
- Sends packet from the buffer at a specified interval

3.3 Experiment

In the experiment, there were mainly two objectives;

- Show how TFC can be provided selectively based on the usage of TFC Request bit (one bit in TrafficClass-field was used) and selective routing.
- Show how TFC mechanisms like layer2-encryption and padding makes it hard to do a traffic flow analysis.

3.3.1 Setup

As seen in figure 9 below, traffic between CC11-CC13 and CC41-CC42 could traverse two paths – either from E6 over E3 or from E6 over E4. Packet-captures were performed on both paths. Between the unclass- and mission-CCs phone-calls were made.

Traffic from mission-CCs was marked by using one bit in the TrafficClass-field, making marked traffic traverse the TFC-provisioned path.

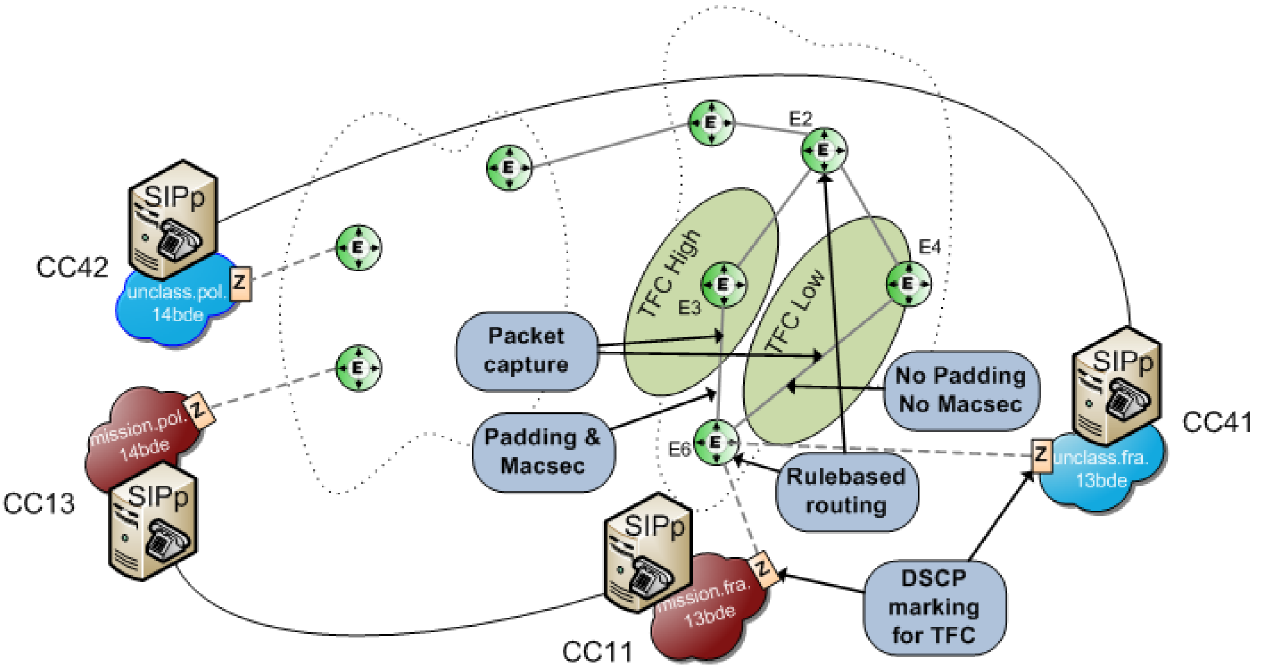
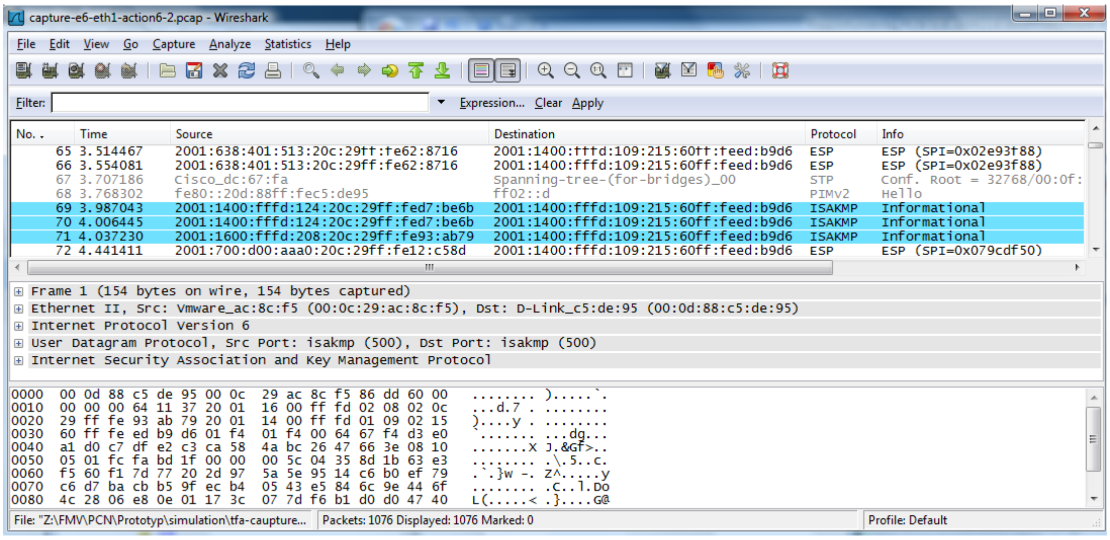


Figure 9: Setup showing TFC-events during the experiment

3.3.2 Results

3.3.2.1 Results, Path without TFC



capture-e6-eth1-action6-2.pcap - Wireshark

File Edit View Go Capture Analyze Statistics Help
Filter:
Expression... Clear Apply

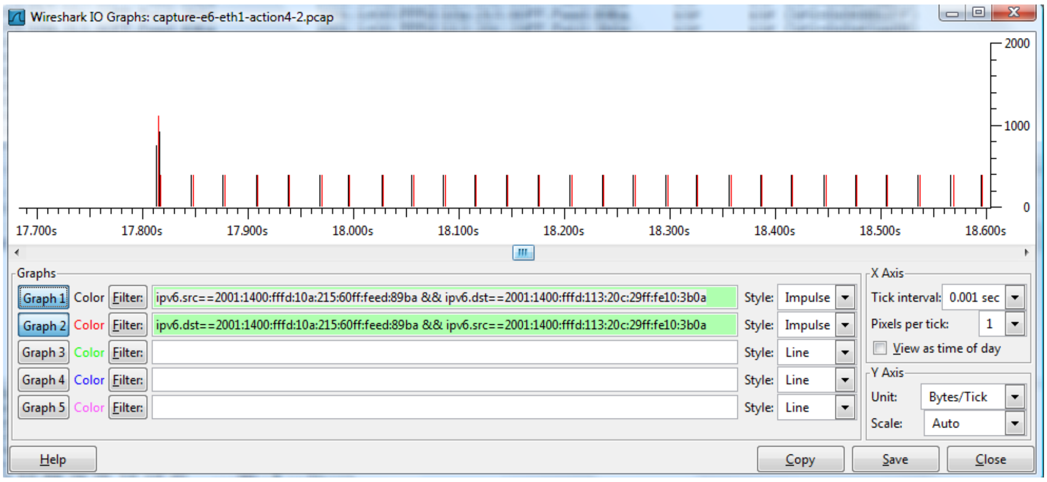
```
No. .  Time      Source                                Destination                           Protocol  Info
    65 3.514467  2001:638:401:513:20c:29ff:fe62:8716   2001:1400:fffd:109:215:60ff:feed:b9d6 ESP       ESP (SPI=0x02e93f88)
    66 3.554081  2001:638:401:513:20c:29ff:fe62:8716   2001:1400:fffd:109:215:60ff:feed:b9d6 ESP       ESP (SPI=0x02e93f88)
    67 3.707186  Cisco_dc:67:fa                        Spanning-tree-(for-bridges)_00        STP       Conf. Root = 32768/00:0f:
    68 3.768302  fe80::20d:88ff:fec5:de95              ff02::d                               PIMv2     Hello
    69 3.987043  2001:1400:fffd:124:20c:29ff:fed7:be6b 2001:1400:fffd:109:215:60ff:feed:b9d6 ISAKMP    Informational
    70 4.006445  2001:1400:fffd:124:20c:29ff:fed7:be6b 2001:1400:fffd:109:215:60ff:feed:b9d6 ISAKMP    Informational
    71 4.037230  2001:1600:fffd:208:20c:29ff:fe93:ab79 2001:1400:fffd:109:215:60ff:feed:b9d6 ISAKMP    Informational
    72 4.441411  2001:700:d00:aaa0:20c:29ff:fe12:c58d  2001:1400:fffd:109:215:60ff:feed:b9d6 ESP       ESP (SPI=0x079cdf50)

⊞ Frame 1 (154 bytes on wire, 154 bytes captured)
⊞ Ethernet II, Src: Vmware_ac:8c:f5 (00:0c:29:ac:8c:f5), Dst: D-Link_c5:de:95 (00:0d:88:c5:de:95)
⊞ Internet Protocol Version 6
⊞ User Datagram Protocol, Src Port: isakmp (500), Dst Port: isakmp (500)
⊞ Internet Security Association and Key Management Protocol

0000  00 0d 88 c5 de 95 00 0c  29 ac 8c f5 86 dd 60 00   ........ ).....`.
0010  00 00 00 64 11 37 20 01  16 00 ff fd 02 08 02 0c   ...d.7 . ........
0020  29 ff fe 93 ab 79 20 01  14 00 ff fd 01 09 02 15   )....y . ........
0030  60 ff fe ed b9 d6 01 f4  01 f4 00 64 67 f4 d3 e0   `....... ...dg...
0040  a1 d0 c7 df e2 c3 ca 58  4a bc 26 47 66 3e 08 10   .......X J.&Gf>..
0050  05 01 fc fa bd 1f 00 00  00 5c 04 35 8d 1b 63 e3   ........ .\.5..c.
0060  f5 60 f1 7d 77 20 2d 97  5a 5e 95 14 c6 b0 ef 79   .`.}w -. Z^.....y
0070  c6 d7 ba cb b5 9f ec b4  05 43 e5 84 6c 9e 44 6f   ........ .C..l.Do
0080  4c 28 06 e8 0e 01 17 3c  07 7d f6 b1 d0 d0 47 40   L(.....< .}....G@

File: "Z:\FMV\PCN\Prototyp\simulation\tfa-caupture...  Packets: 1076 Displayed: 1076 Marked: 0  Profile: Default
```

Figure 10: Packet-capture from E4-E6

In the packet-capture shown in figure 10, no traffic between the MissionCC-peers can be observed. Since there was a working phone call between the MissionCCs the traffic has to have traversed the other path (E3-E6). However, the packet-capture does include a lot of ESP-packets between several peers thus providing good material for doing a traffic flow analysis.

One flows stand out as more active than others; 2001:1400:fffd:10a[…] – 2001:1400:fffd:113:[…] (unclass-CCs).

Figure 11 below shows this flow from another perspective showing when packets are sent and how big they are. Red and black are representing traffic in different directions.



Wireshark IO Graphs: capture-e6-eth1-action4-2.pcap

Graph 1 Color Filter: ipv6.src==2001:1400:fffd:10a:215:60ff:feed:89ba && ipv6.dst==2001:1400:fffd:113:20c:29ff:fe10:3b0a Style: Impulse

Graph 2 Color Filter: ipv6.dst==2001:1400:fffd:10a:215:60ff:feed:89ba && ipv6.src==2001:1400:fffd:113:20c:29ff:fe10:3b0a Style: Impulse

Graph 3 Color Filter: Style: Line

Graph 4 Color Filter: Style: Line

Graph 5 Color Filter: Style: Line

Tick interval: 0.001 sec

Pixels per tick: 1

☐ View as time of day

Unit: Bytes/Tick

Scale: Auto

Figure 11: IO graph on traffic between unclass-CCs

After approximately 18 seconds from starting the packet-capture, two streams in either direction can be observed between unclass-CCs lasting about 8 seconds. The interval between the packets lies steady at about 30ms, and the size of ESP payload is 332 bytes.

An RTP-stream at 64kbps (G711) generates approximately 33 packets per second which equals an interval of 30ms. When looking at the size of an RTP-packet [15], the IPv6 header and payload sum up at 300 bytes.

Comparing this to the captured ESP packets shows similarities in both interval and size, so let’s dig a little bit deeper.

Looking at ESP [16] in figure 12, there are a few fixed-size headers and some variable.

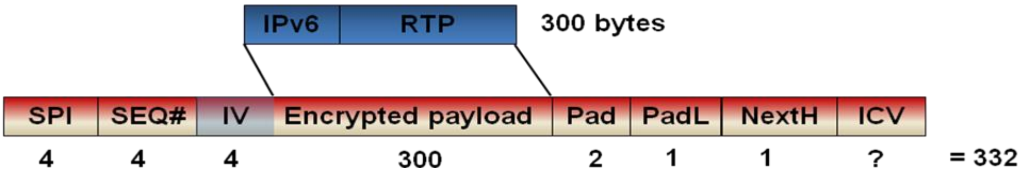
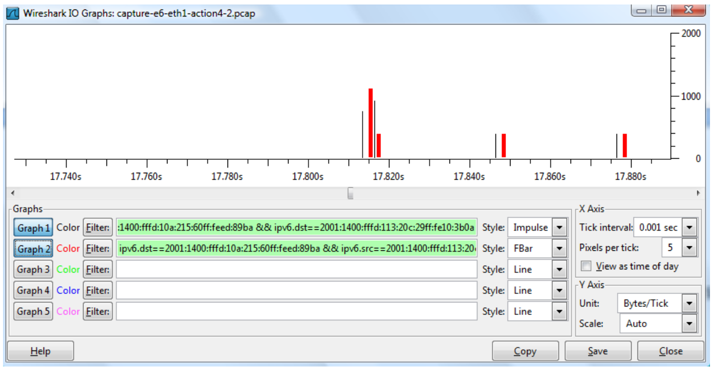


Figure 12: ESP headers and payload

The fixed-size headers are: SPI (4 bytes), sequence (4 bytes), pad length (1 byte) and next header (1 byte). The variable headers are: padding (a variable length, making payload-size=n*blockciphersize) and ICV (depending on algorithm – 16 bytes for MD5 and 20 bytes for SHA-1).

Starting with an RTP-packet of 300 bytes, we first add 4 bytes as there probably is an IV in the ESP payload, then we pad with 2 bytes for the ESP payload-data (including pad-length and next-header) to stop at a 32-bit-boundary. Then we add the SPI and sequence, now we have 316 bytes. Finally, adding an ICV of 16 bytes leaves us at a total of 332 bytes which matches the actual size of the ESP payload from the packet-capture. So by looking at sizes and intervals of packets we can make an assumption that this is really RTP, further analysis showed that MD5 probably was used as the hashing algorithm creating the ICV.

When zooming in on the beginning of the call, shown in figure 13, the first that can be seen is a black impulse followed by red-black-red and then the streams start. A good guess would be that black (being CC11 = 2001:1400:fffd:10a…) initiated the call.


Wireshark IO Graphs: capture-e6-eth1-action4-2.pcap

Graphs
Graph 1 Color Filter: :1400:fffd:10a:215:60ff:feed:89ba && ipv6.dst==2001:1400:fffd:113:20c:29ff:fe10:3b0a Style: Impulse
Graph 2 Color Filter: ipv6.dst==2001:1400:fffd:10a:215:60ff:feed:89ba && ipv6.src==2001:1400:fffd:113:20 Style: FBar
Graph 3 Color Filter: Style: Line
Graph 4 Color Filter: Style: Line
Graph 5 Color Filter: Style: Line

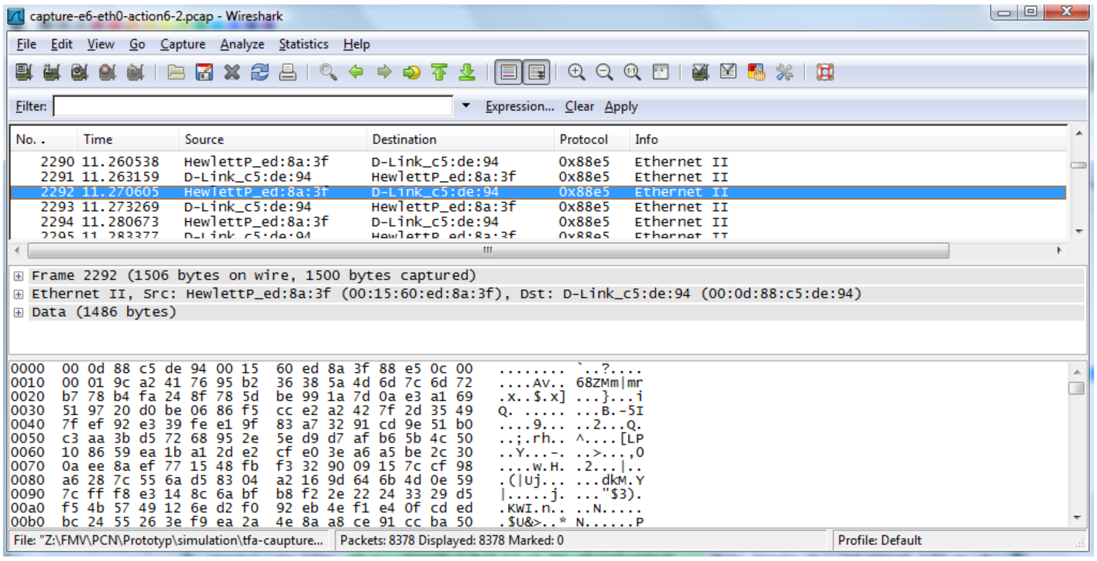
X Axis
Tick interval: 0.001 sec
Pixels per tick: 5
□ View as time of day

Y Axis
Unit: Bytes/Tick
Scale: Auto

Figure 13: IO graph zooming in on traffic between unclass-CCs

If there were more flows – phone-calls to analyze, a chain-of-command could possibly be visualized just by watching who normally initiates the calls.

3.3.2.2 Results, Path with TFC


capture-e6-eth0-action6-2.pcap - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter:
Expression... Clear Apply

```
No. .  Time       Source             Destination        Protocol  Info
  2290 11.260538  HewlettP_ed:8a:3f  D-Link_c5:de:94    0x88e5    Ethernet II
  2291 11.263159  D-Link_c5:de:94    HewlettP_ed:8a:3f  0x88e5    Ethernet II
  2292 11.270605  HewlettP_ed:8a:3f  D-Link_c5:de:94    0x88e5    Ethernet II
  2293 11.273269  D-Link_c5:de:94    HewlettP_ed:8a:3f  0x88e5    Ethernet II
  2294 11.280673  HewlettP_ed:8a:3f  D-Link_c5:de:94    0x88e5    Ethernet II
  2295 11.283377  D-Link_c5:de:94    HewlettP_ed:8a:3f  0x88e5    Ethernet II
```

```
⊞ Frame 2292 (1506 bytes on wire, 1500 bytes captured)
⊞ Ethernet II, Src: HewlettP_ed:8a:3f (00:15:60:ed:8a:3f), Dst: D-Link_c5:de:94 (00:0d:88:c5:de:94)
⊞ Data (1486 bytes)
```

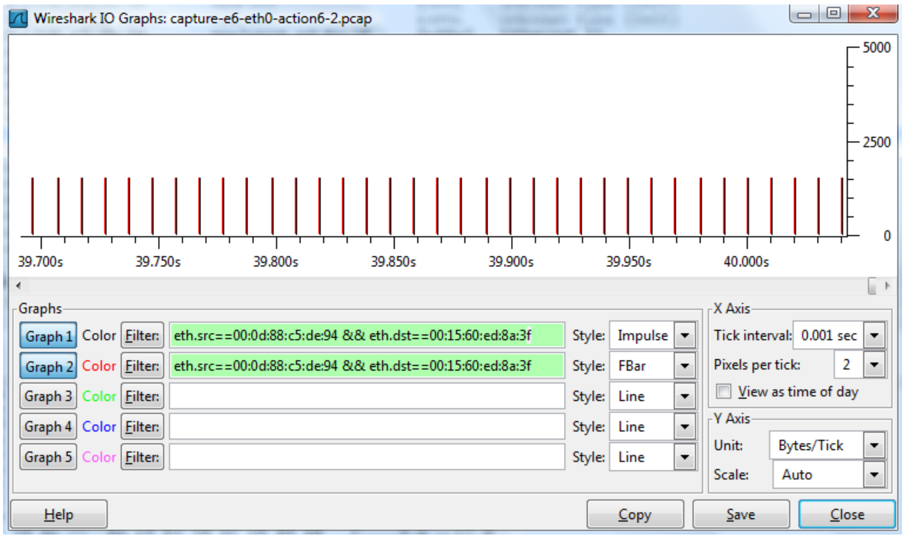
```
0000  00 0d 88 c5 de 94 00 15  60 ed 8a 3f 88 e5 0c 00   ........ `..?....
0010  00 01 9c a2 41 76 95 b2  36 38 5a 4d 6d 7c 6d 72   ....Av.. 68ZMm|mr
0020  b7 78 b4 fa 24 8f 78 5d  be 99 1a 7d 0a e3 a1 69   .x..$.x] ...}...i
0030  51 97 20 d0 be 06 86 f5  cc e2 a2 42 7f 2d 35 49   Q. ..... ...B.-5I
0040  7f ef 92 e3 39 fe e1 9f  83 a7 32 91 cd 9e 51 b0   ....9... ..2...Q.
0050  c3 aa 3b d5 72 68 95 2e  5e d9 d7 af b6 5b 4c 50   ..;.rh.. ^....[LP
0060  10 86 59 ea 1b a1 2d e2  cf e0 3e a6 a5 be 2c 30   ..Y...-. ..>...,0
0070  0a ee 8a ef 77 15 48 fb  f3 32 90 09 15 7c cf 98   ....w.H. .2...|..
0080  a6 28 7c 55 6a d5 83 04  a2 16 9d 64 6b 4d 0e 59   .(|Uj... ...dkM.Y
0090  7c ff f8 e3 14 8c 6a bf  b8 f2 2e 22 24 33 29 d5   |.....j. ..."$3).
00a0  f5 4b 57 49 12 6e d2 f0  92 eb 4e f1 e4 0f cd ed   .KWI.n.. ..N.....
00b0  bc 24 55 26 3e f9 ea 2a  4e 8a a8 ce 91 cc ba 50   .$U&>..* N......P
```

File: "Z:\FMV\PCN\Prototyp\simulation\tfa-caupture... Packets: 8378 Displayed: 8378 Marked: 0 Profile: Default

Figure 14: Packet-capture from E3-E6

The packet-capture in figure 14 shows only a lot of encrypted traffic (MACSec) between two mac-addresses.

When looking at interval and size in figure 15, all frames are sent with same interval (10ms) and all are of same size (1500 bytes). This will make it very hard to gather interesting information regarding traffic-flows.



Graph 1 Color Filter: eth.src==00:0d:88:c5:de:94 && eth.dst==00:15:60:ed:8a:3f Style: Impulse
Graph 2 Color Filter: eth.src==00:0d:88:c5:de:94 && eth.dst==00:15:60:ed:8a:3f Style: FBar
Graph 3 Color Filter: Style: Line
Graph 4 Color Filter: Style: Line
Graph 5 Color Filter: Style: Line

Tick interval: 0.001 sec
Pixels per tick: 2
□ View as time of day
Unit: Bytes/Tick
Scale: Auto

Figure 15: IO graph on traffic between E3 and E6

3.3.3 Conclusion

In a network without any TFC-mechanisms, it's possible just by watching encrypted flows between two endpoints, to gather important information revealing protocols in use and possibly a chain-of-command. Knowing protocols in use will make cryptanalysis a bit easier since both plain-text, although partially, and cipher-text is given.

On the other hand, on a link provisioned with mechanisms like layer2-encryption and padding, it will be very hard to get any substantial value at all from a traffic flow analysis.

4.0 WITHSTANDING DIRECTED ATTACKS

When a traditional network is being subject to a DoS(Denial of Service)-attack there are very few mechanisms to ensure that the impact is as small as possible. An unauthorized source may be able to inject packets at a rate which equals the bandwidth of its own link, which can heavily impact the network infrastructure and in turn the users.

4.1 PCN Concept

In PCN, only authorized sources are permitted to use the network for transport. This is achieved by only granting authenticated and authorized users access to the network. Further, utilizing per-packet-protection, injection of unauthorized traffic reaching the network is not possible.

All connections on PCN-1 and PCN-2 are subject to policing, traffic exceeding the previously agreed SLA(Service Level Agreement)-rate will be dropped.

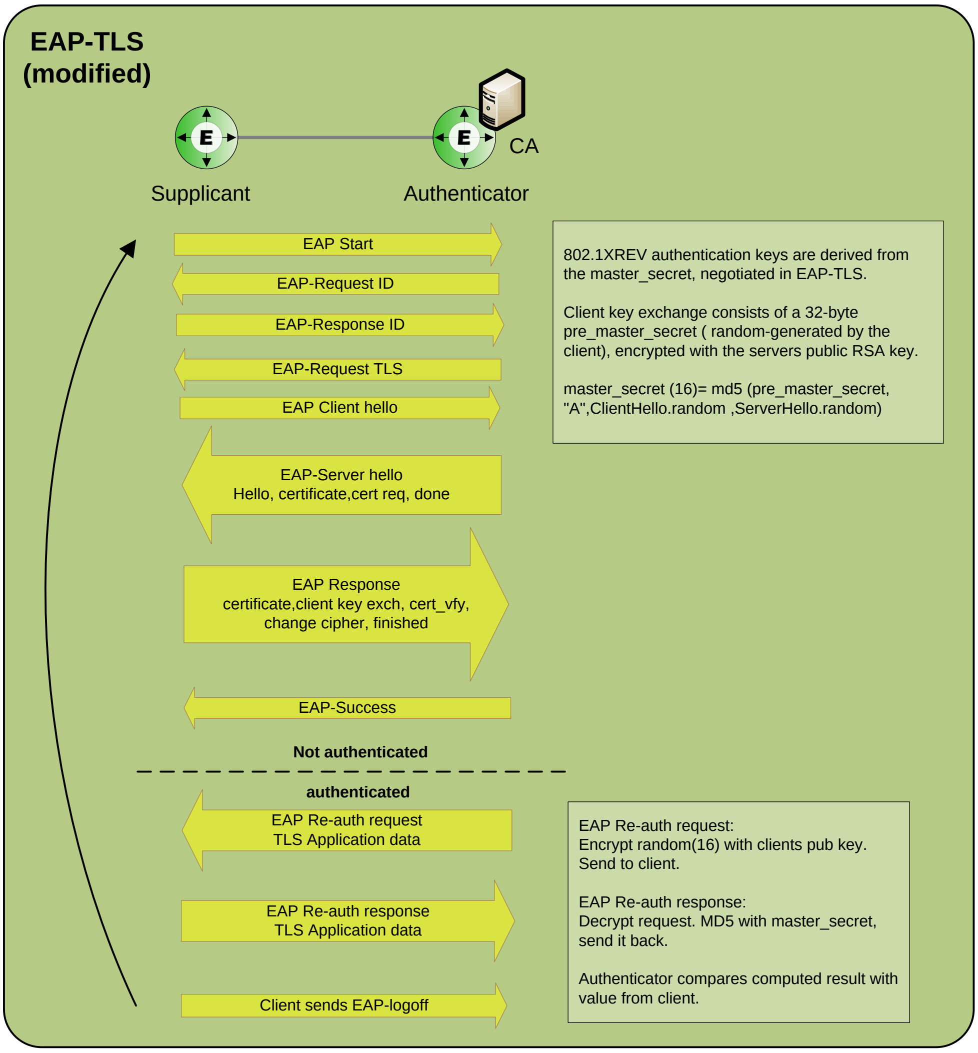
Every PCS includes a cyber defence-capability, enabling detection of, and timeliness reaction to a DoS-attack [17].

4.2 Implementation

The functionality that is implemented in the prototype, only allows authenticated and authorized users to make use of the network. The E-nodes will do policing at a traffic rate agreed upon in a SLA-negotiation, thus stopping unlimited rates. Also, the E-nodes will monitor the traffic rates and if they exceed the SLA for a certain period of time, the interface in subject will be put in an unauthorized state.

4.2.1 Authentication

The authentication is done on layer2, using a modified subset of EAP-TLS (see figure 16). Re-authentication is done in a proprietary way with configurable timers. After a successful authentication EAP-TLS has negotiated a shared secret, this secret is later on used for authentication in the setup of MACSec.



802.1XREV authentication keys are derived from the master_secret, negotiated in EAP-TLS.

Client key exchange consists of a 32-byte pre_master_secret (random-generated by the client), encrypted with the servers public RSA key.

master_secret (16)= md5 (pre_master_secret, "A",ClientHello.random ,ServerHello.random)

EAP Re-auth request:
Encrypt random(16) with clients pub key.
Send to client.

EAP Re-auth response:
Decrypt request. MD5 with master_secret, send it back.

Authenticator compares computed result with value from client.

Figure 16: Overview of slightly modified EAP-TLS

To be able to perform authentication, the following credentials need to be in place: own certificate, own private key, CA certificate, trusted CA certificates (being able to connect to devices from with credentials from other CAs) and CRLs installed

4.2.2 Authorization

The authorization is a very basic implementation using textfiles residing locally on E-nodes and Z-devices. There are two files, whitelist and blacklist;

Whitelist contains text-rows with parts of certificate Subject CN (if connecting to any Swedish node is ok, something like “se.pcn” could be added – e.g. enabling a connection to e23.se.pcn).

Blacklist contains text-rows with parts of or the entire certificate Subject CN (if a CC has been compromised and the certificate is not yet revoked, a line like z23.se.pcn would reject the connection from a CC with z23.se.pcn in its Subject CN).

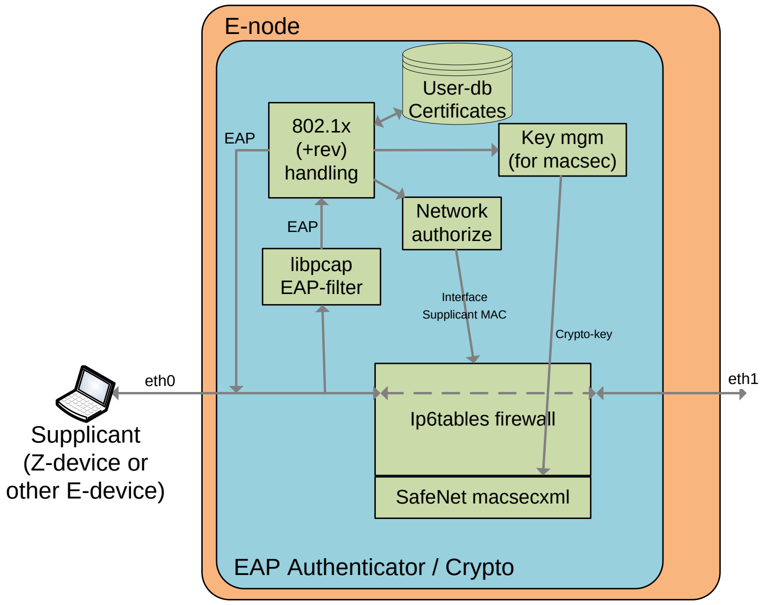
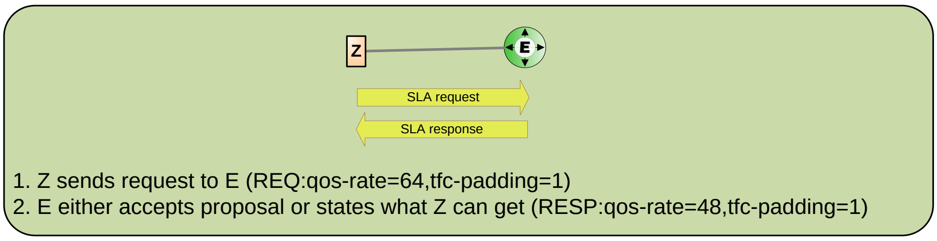


Figure 17: Implementation of authentication/authorization and MACSec

As shown in figure 17, when a device successfully has authenticated, authorization is checked and a decision is made to open or to close the ip6tables firewall for traffic on the interface in subject.

4.2.3 SLA Negotiation and Policing

This is implemented on the PCN-2 interface with the E-node being a server and the Z-device being a client (figure 18). After receiving an address, the Z-device starts the SLA negotiation by stating what bandwidth it wants. The E-node answers with a per-interface-preconfigured value, which results in the Z-device and E-node doing shaping.



1. Z sends request to E (REQ:qos-rate=64,tfc-padding=1)
2. E either accepts proposal or states what Z can get (RESP:qos-rate=48,tfc-padding=1)

Figure 18: Implementation of SLA negotiation

4.2.4 Traffic Rate Monitoring

After negotiating a SLA, the E-node will monitor the traffic-rate. Since the Z-device is shaping its outgoing traffic, exceeding the agreed rate shouldn’t occur. However, if the E-node notices that the Z-device exceeds the rate agreed in the SLA for a certain amount of time, the E-node will add the Z-device to its blacklist. This will lead to disconnection of the Z-device.

4.2.5 Redundancy Routing

Having more than one path between two PCSs is of course good for redundancy. If the traffic rate monitoring, mentioned in previous chapter, is shutting down an interface due to a DoS-attack on a link, routing protocols will be aware of this and make traffic traverse a redundant path. In the prototype this was configured using OSPF and BGP.

4.3 Experiment

In the experiment, there were three objectives;

- Show how a DoS-attack, both on a user interface and a network link, has great impact throughout the network in a non-PCN scenario.
- Show how a DoS-attack in a PCN scenario, both on a user interface and a network link, has little or no impact on the network.
- Show that a compromised CC is limited in the effect it can have by the agreed SLA. Also show that by revoking a CCs credential, the compromised CC will not be able to connect to the PCore.

4.3.1 DoS on PCN-1

4.3.1.1 Setup, DoS on PCN-1

The DoS-attack as shown in figure 19 was performed on the PCN-1 link between E7 and E15.

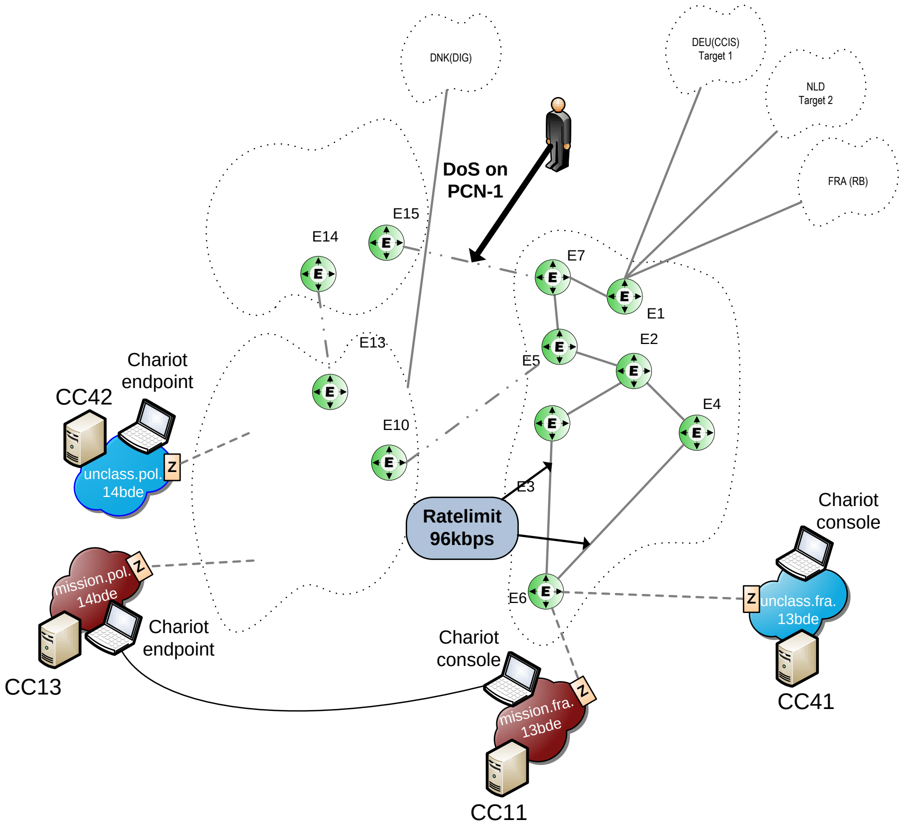


Figure 19: DoS-attack on PCN-1, source and targets

The attack was first performed without PCN-functionality and then with PCN-functionality to be able to note any differences.

4.3.1.2 Results, DoS on PCN-1

Impact on PCSs and CCs (seen from management within PCS FRA and CCs) is shown in figure 20 below.

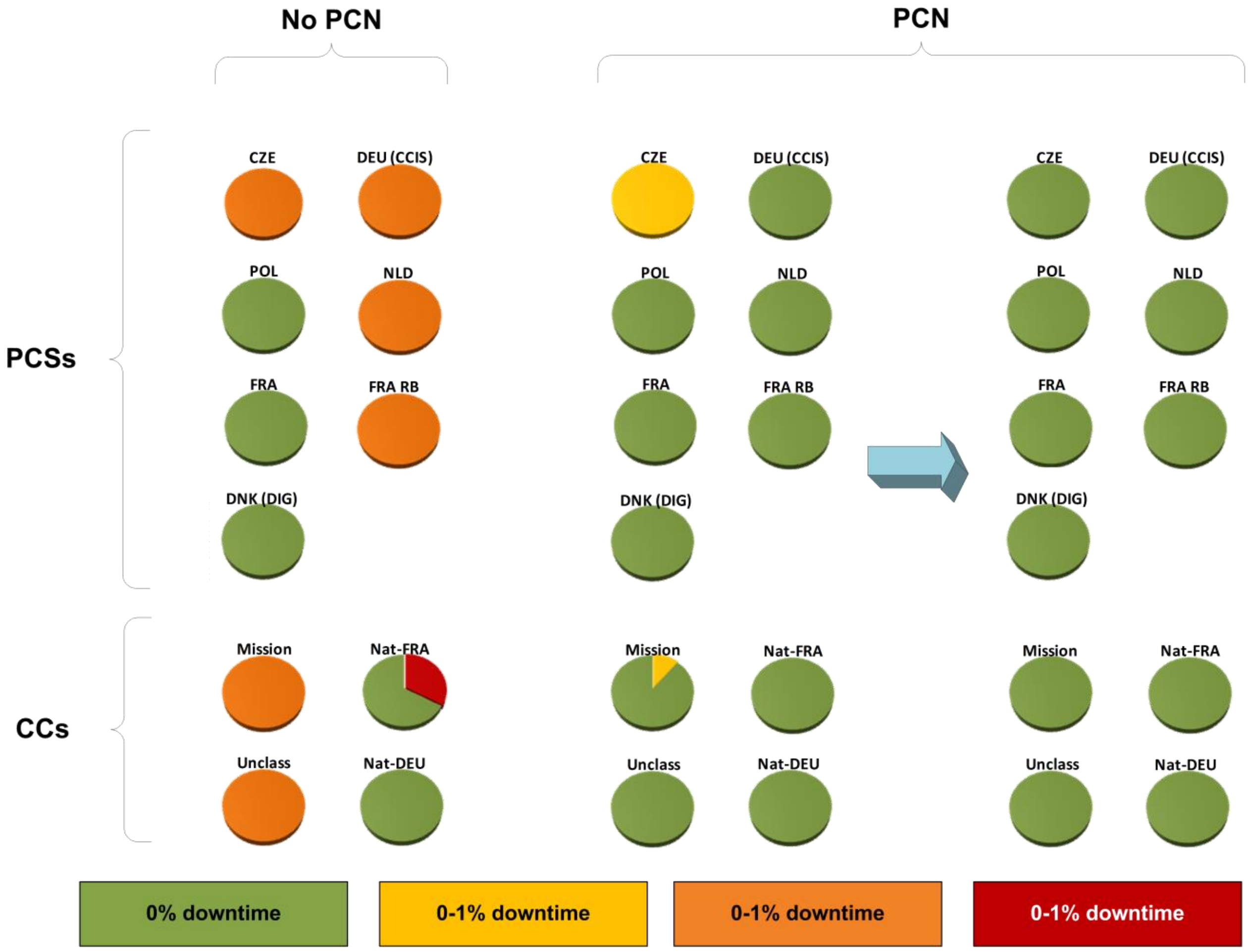


Figure 20: DoS-attacks on PCN-1, PCS, CC impact comparison.

In the run with PCN-functionality, ICMPv6 was manually added as allowed traffic from the DoS-machine. Without this, neighbour discovery wouldn’t have worked and the impact from the DoS would have been none.

When traffic monitoring in an E-node detected a traffic-rate that exceeded the SLA for a while, the E-node shut down the interface towards the DoS-machine. Routing-protocols noticed this and made a redundant path active, thus making the PCore go from a state with little impact to a state with no impact at all.

In figure 21 and 22, the impact on traffic between CCs is shown. The DoS-attack was based on udp and the throughput-tests were based on tcp, which could explain the small impact. During the run with PCN-functionality, mission-CCs are getting more bandwidth due to prioritization done in the PCore.

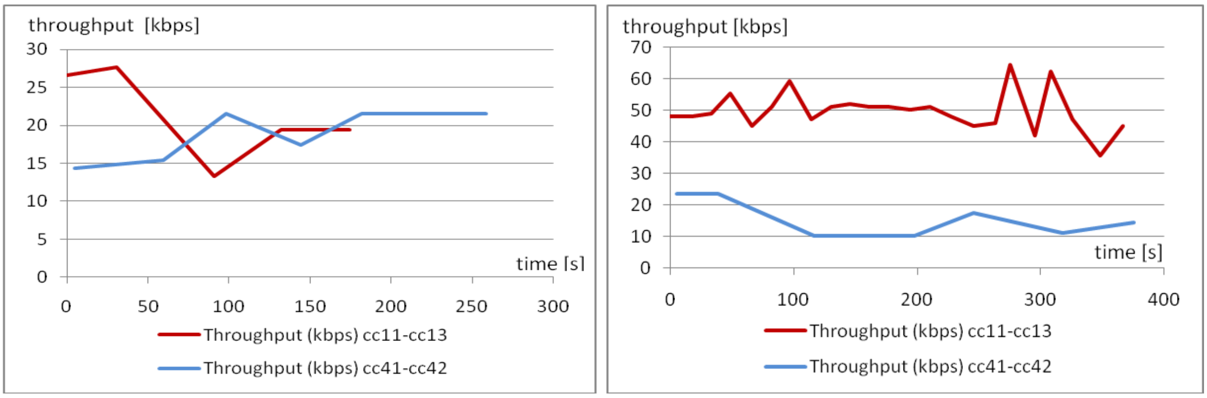


Figure 21 and 22: Throughput for "no pcn"-run (left) and "pcn"-run (right)

4.3.2 DoS on PCN-2

4.3.2.1 Setup, DoS on PCN-2

The attacks on PCN-2 are divided in three stages as shown in figure 23.

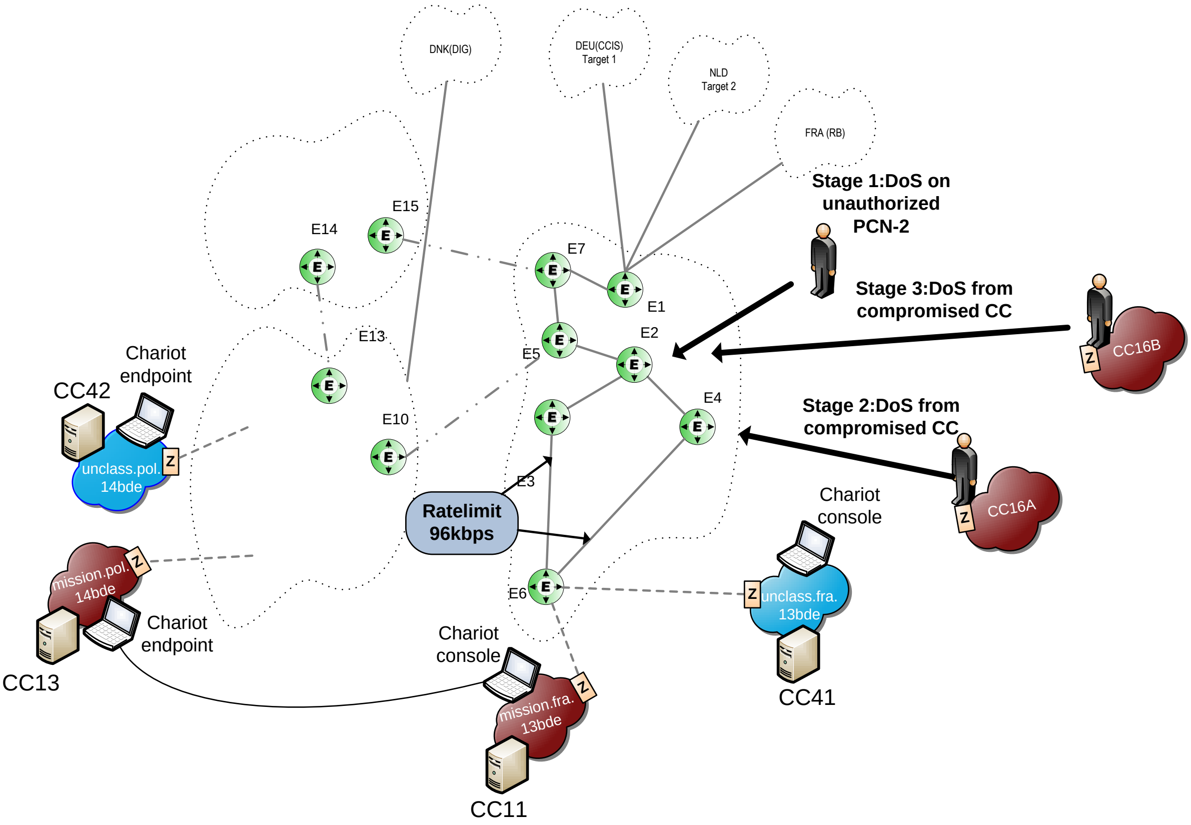


Figure 23: DoS-attacks from a CC

- In stage 1, a DoS-attack was performed, both with and without PCN-functionality, from a machine connected to E2.

- In stage 2 and 3, PCN-functionality was enabled. Stage 2 involves a non-authorized CC doing a DoS-attack.

- In stage 3, a compromised CC is doing a DoS-attack. When the connection is shut down, due to traffic rate exceeding SLA the CC relocates. A decision (out-of-band) is made that the CC is compromised and the credentials of the CC are revoked. The CC tries to connect to another E-node.

4.3.2.2 Results, DoS on PCN-2

Impact on PCSs and CCs (seen from management within PCS FRA and CCs) is shown in figure 24 below.

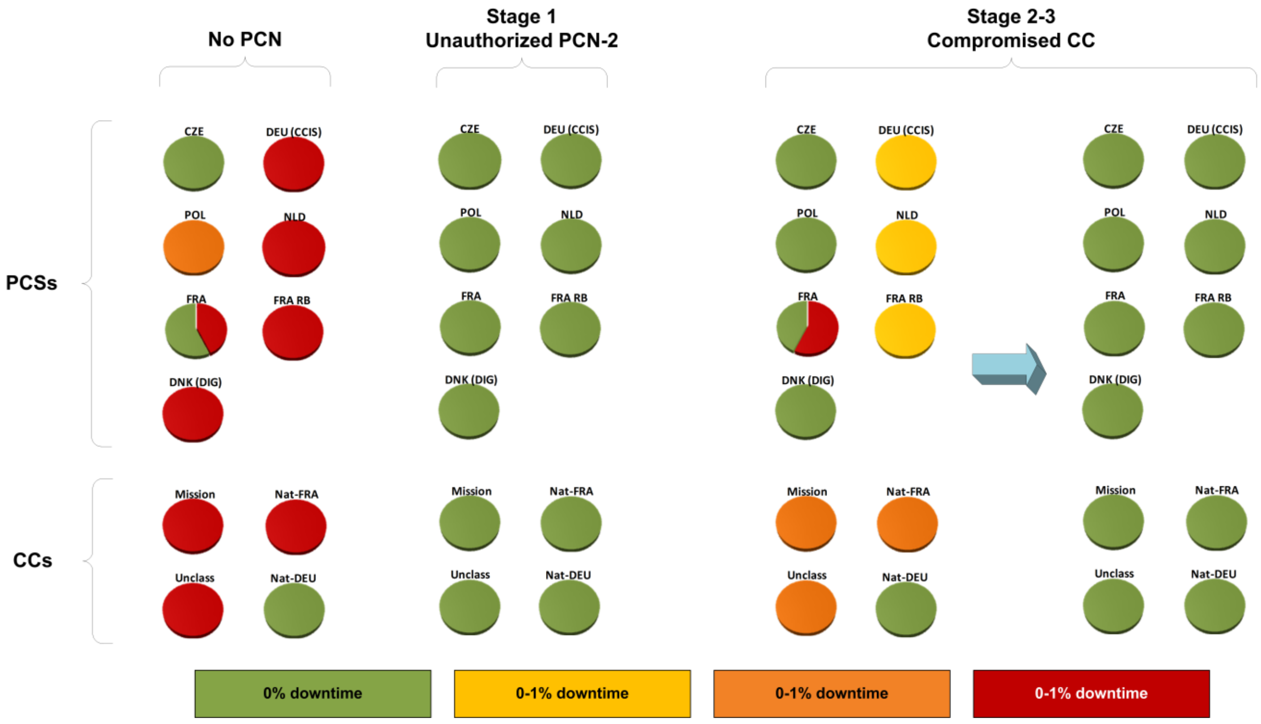


Figure 24: DoS-attacks on PCN-2, PCS&CC impact comparison

The impact for both PCSs and CCs during the DoS without PCN-functionality was very high. With a DoS-attack from an unauthorized PCN-2, the impact was none.

In the run with a compromised CC, the E-node policed traffic at a pre-agreed SLA-rate thus reducing the impact. Then, when noticing traffic-rate exceeding the SLA for a period of time, the E-node shut down the interface towards the CC. This made the PCore go from a state with little impact to a state with no impact at all.

In figure 25 and 26, the impact on traffic between CCs is shown. The DoS-attack was based on udp and the throughput-tests were based on tcp, which could explain the small impact. During the run with PCN-functionality, mission-CCs are getting more bandwidth due to prioritization done in the PCore.

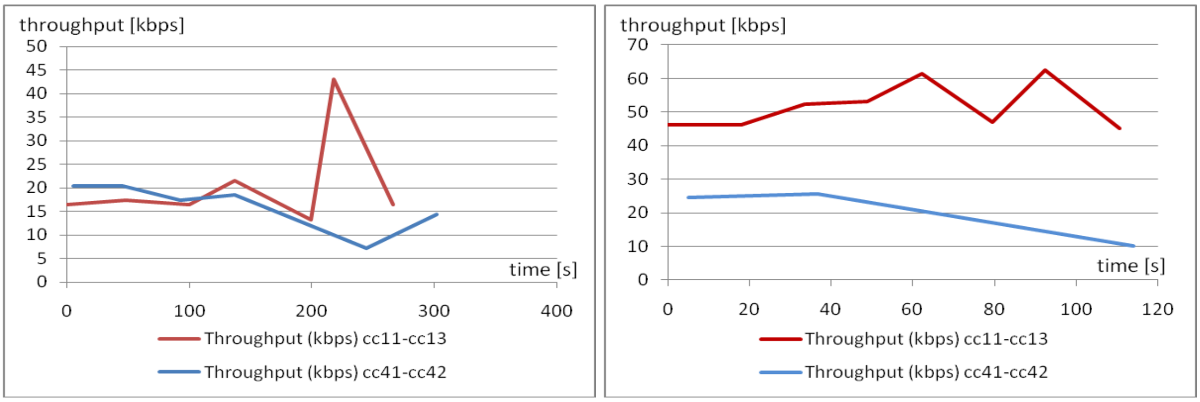


Figure 25,26: Throughput for "no-pcn"-run (left) and "pcn"-run (right)

In stage 3, when the CC after being shut down tried to connect to another E-node the authentication failed due to a revocation of the CC certificate.

4.3.3 Conclusions

The attacks on PCN-1 interface show that without PCN-functionality the impact was low on both PCore and CC-communication and with PCN-functionality the impact was very low. The impact would probably been higher if directing the attack at other targets.

When DoS was performed with PCN-functionality, there was no impact at all due to the fact that the DoS worked on the IP-layer and since packets came from an unauthorized source, neighbor discovery didn't work. Hence the DoS-source couldn't get the mac-address of its default gateway. To circumvent this, in a way reflecting a scenario with injection, icmpv6 was allowed. This resulted in a big load on the link, but still no DoS-packets inside any PCS, since the packets were coming from an unauthorized source (in this case mac-address). To further disqualify unwanted traffic - avoiding address-spoofing, per-packet-protection could be utilized.

One thing to notice here is the capability of a PCore to recover during an attack and get back to a normal almost unaffected state. This is done by monitoring the traffic rate, making the connecting E-node in the other PCS unauthorized if exceeding the agreed SLA for a specified time (here 10 seconds). When routing protocols (BGP) noticed that the connection to the neighbour was down (configured to 60 seconds), another route from the disconnected PCS came in effect and the communication to the lost PCS worked again. The policy to shutdown connections may not always be the preferred alternative, but proves the dynamics and strength with PCN.

The attacks on PCN-2 interface show that without PCN-functionality, the impact is high on both PCore- and CC-communication.

An attack on an unauthorized PCN-2 has no effect at all and an attack from a compromised CC has a very limited impact. One thing to notice, as well as in the attacks on PCN-2, is the capability of the PCore to recover during an attack and get back to a normal unaffected state. This is done by monitoring the traffic rate, making the CC unauthorized if exceeding the agreed SLA for a specified time (here 120 seconds). Since the CC was considered compromised its PCN-credentials were revoked, thereby not allowing the CC to connect at any other point in the PCore.

The throughput-tests (TCP-based) during the attacks on PCN-1 and PCN-2, show no big impact from the DoS-attacks, it probably would have been more interesting doing a VoIP-jitter test. During the DoS-runs with PCN-functionality QoS was used, this can be seen in both throughput-tests where the Mission-CCs are getting more bandwidth than the Unlcass-CCs.

5.0 CONCLUSIONS

The prototype shows that the concept of PCN has been successfully validated in terms of mobility, traffic flow confidentiality and protection against directed attacks.

The need for flexibility to facilitate relocation of coloured clouds is satisfied by using auto-configuration and a mechanism for peer discovery.

Provision of traffic flow confidentiality proves to be a countermeasure against analysis of encrypted traffic flows between coloured clouds trying to gather information of the communication.

When it comes to the capability to withstand directed attacks, a network built on PCN-principles not only reduces but could even eliminate the impact of a DoS-attack. This is be achieved by utilizing policing on incoming traffic and only allowing authenticated and authorized sources to send traffic to the network. When combining these two measures with a cyber defence-capability, enabling detection and reaction to attacks, the impact is even more reduced and enables the network to automatically recover during an attack.

6.0 REFERENCES

[1] Quagga Routing Suite, http://www.quagga.net

[2] R. Coltun, D. Ferguson, J. Moy, "OSPF for IPv6", IETF RFC2740, December 1999

[3] Y. Rekhter, T. Li, S. Hares, "A Border Gateway Protocol 4 (BGP-4)", IETF RFC 4271, January 2006

[4] MRD6, an IPv6 Multicast Router, http://fivebits.net/proj/mrd6/

[5] B. Fenner, M. Handley, H. Holbrook, I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", IETF RFC 4601, August 2006

[6] R. Vida, L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", IETF RFC 3810, June 2004

[7] D. Thaler, "Border Gateway Multicast Protocol (BGMP): Protocol Specification", IETF RFC 3913, September 2004

[8] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, H. Levkowetz,"Extensible Authentication Protocol (EAP)", IETF RFC 3748, March 2008

[9] D. Simon, B. Aboba, R. Hurst, "The EAP-TLS Authentication Protocol",

[10] T. Dierks, C. Allen, "The TLS Protocol Version 1.0", IETF RFC 2246, January 1999

[11] S. Thomson, T. Narten,"IPv6 Stateless Address Autoconfiguration", IETF RFC 2462, December 1998

[12] T. Aurisch, T. Ginzler, P. Martini, R. Ogden, T. Tran, H. Seifert, "Automatic multicast IPsec by using a proactive IPsec discovery protocol and a group key management", Journal of telecommunications and information technology 2/2008

[13] T. Aurisch, D. Dahlberg, M. Lies, P. Sevenich, "An approach towards traffic flow confidentiality in military IPsec protected networks", MILITARY COMMUNICATIONS AND INFORMATION SYSTEMS CONFERENCE 2009

[14] G. Hallingstad,F. Micevski Scharf, "PROVISION OF MULTIPLE LEVELS OF TRAFFIC FLOW CONFIDENTIALITY-SERVICE IN PROTECTED CORE NETWORKS", RTO IA Symposium 2008

[15] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", IETF RFC 3550, July 2003

[16] S. Kent, "IP Encapsulating Security Payload (ESP)", IETF RFC 4303, December 2005

[17] S. Oudkerk, C. Verkoelen, “Policy Based Network Management in Protected Core Networking”, RTO IA Symposium 2008

[18] Eclipse IDE , http://www.eclipse.org/

[19] VMWare, http://www.vmware.com/

[20] Ubuntu, http://www.ubuntu.com/

[21] SafeNet, http://www.safenet-inc.com/

7.0 ABBREVIATIONS

BGP	Border Gateway Protocol
CA	Certificate authority
CC	Coloured Cloud
CRL	Certificate Revocation List
CWID	Coalition Warrior Interoperability Demonstration
DoS	Denial of Service
EAP	Extensible Authentication Protocol
ESP	Encapsulating Security Payload
ICMPv6	Internet Control Message Protocol Version 6
ICV	Integrity Check Value
IGMP	Internet Group Management Protocol
IKE	Internet Key Exchange
IP	Internet Protocol
IPv6	Internet Protocol version 6
IPSec	Internet Protocol Security
IV	Initialization Vector
LCC	Land Component Command
MLD	Multicast Listener Discovery
OSPF	Open Shortest Path First
PCN	Protected Core Network
PCS	Protected Core Segment
PIM	Protocol-Independent Multicast
RTO	NATO Research and Technology Organisation
RTP	Real-time Transport Protocol
SLA	Service Level Agreement
TFC	Traffic Flow Confidentiality
TLS	Transport Layer Security